CONSTRUCTION OF MAXIMAL DYNAMIC FLOWS IN NETWORKS

L. R. Ford, Jr., and D. R. Fulkerson

25

P-1079

May 7, 1957

13p

The RAND Corporation
1700 MAIN ST. • SANTA MONICA • CALIFORNIA

SUMMARY

An algorithm for solving the problem of finding a maximal dynamic flow through a network is described. No proofs are given.

CONSTRUCTION OF MAXIMAL DYNAMIC FLOWS IN NETWORKS

L. R. Ford, Jr., and D. R. Fulkerson

1. Introduction. In this note we describe briefly an easy and efficient algorithm for solving the following problem. Suppose given a network (linear graph) in which each link has associated with it two positive integers, one a commodity flow capacity, the other a traversal time. Assuming that some node of the network is a source for the commodity, another a sink, and the remaining may either transship the commodity immediately on receipt or hold for later shipment, what is the maximal amount that can be shipped from source to sink in any given number of time periods?

For example, in the toy network of Figure A, P_0 is the source, P_3 the sink, and the capacities of the links in terms of flow per unit time are the first numbers of the pairs, the traversal times the second numbers. How many units of commodity flow can reach P_3 from P_0 in five time periods, say, and what is a flow pattern which achieves this?

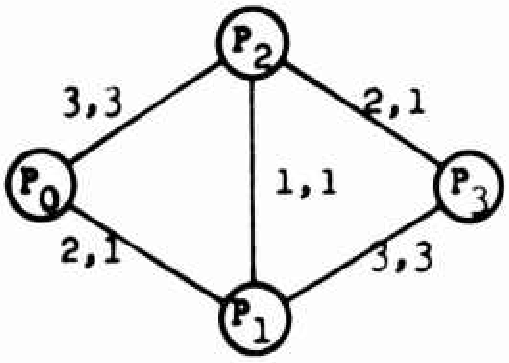

Figure A

A simpler problem related to the dynamic one just described is the maximal static flow problem, i.e., assuming a steady state condition, find a maximal flow from source to sink in a capacitated network. The labeling process [7] provides a simple method of solution for this latter problem; a variation of this process is also used as a subroutine in the algorithm for the dynamic problem.

The construction of an optimal dynamic flow for a given number T of time periods first produces a static flow in the network which has certain prescribed properties. The dynamic solution is then obtained from the static solution by decomposing the latter into "chain flows," starting each chain flow at time zero, and continuing each so long as there is enough time left in the T periods for the flow along the chain to arrive at the sink. Thus, for the network of Figure A, the static flow produced by the algorithm for T = 5 is shown in Figure B below. This flow decomposes into a flow of two units along each of the chains $P_0P_1P_3$, $P_0P_2P_3$. Since each of these chains has a total traversal time of four, each can be used twice in the five periods, hence eight units of the commodity reach P_3 from P_0 in the total time interval in question. The fact that a maximal dynamic flow of this particularly simple kind exists for all T seems to us rather remarkable.

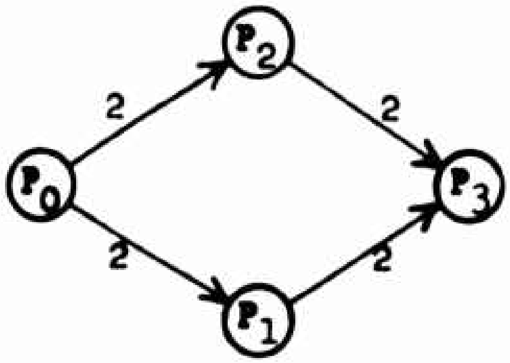

Figure B

No proofs will be given in this paper. These may be found in [9].

2. Definitions and Heuristic Discussion. Let $P_0, P_1, \ldots, P_n$ be the nodes of a network N, and denote by P_iP_j the link from P_i to P_j (in that order). Associated with each link P_iP_j present in N are two positive integers, c_{ij} and t_{ij}, its capacity and traversal time, respectively. (In the network of Figure A the capacities and times are assumed symmetric, $c_{ij} = c_{ji}$ and $t_{ij} = t_{ji}$, but this is not necessary in the algorithm.) The notation is taken so that P_0 is the source, P_n the sink.

A static flow in N is a collection of numbers (x_{ij}), one for each link P_iP_j of N, such that

$$(1) \qquad \sum_{j=0}^{n} (x_{ij} - x_{ji}) = 0 \qquad (i = 1, \ldots, n-1)$$

$$0 \leq x_{ij} \leq c_{ij} .$$

(It is assumed in (1), and elsewhere, that a variable x_{ij} appears only if P_iP_j is a link of N.) The value of the static flow (x_{ij}) is the total amount leaving P_0, $\sum_j (x_{0j} - x_{j0})$. Thus the static maximal flow problem may be expressed as

$$(2) \qquad \text{maximize} \sum_{j=1}^{n} (x_{0j} - x_{j0})$$

subject to the equations and inequalities (1). In this form, the static problem can be solved by the algorithm given in [7].

An alternative formulation of the static problem appears if one focuses attention on chains in N which lead from P_0 to P_n [6]. Thus, let $C_1, \ldots, C_m$ be a listing of chains from P_0 to P_n, $L_1, \ldots, L_p$ a listing of links of N, and suppose $x_1, \ldots, x_m$ represent the amounts of flow along the chains. If (a_{rs}) is the p by m incidence matrix of links vs. chains, then the problem is to

$$(3) \qquad \text{maximize} \sum_{k=1}^{m} x_k$$

subject to the restrictions

$$(4) \qquad \sum_{k=1}^{m} a_{rk}x_k \leq c_r$$

$$x_k \geq 0$$

where c_r is the capacity of the r-th link.

The chain formulation of the static problem does not seem to be fruitful in a computational sense directly. However, looking at the problem in this form provides some insight into the dynamic problem. Suppose one knew a priori that there were a maximal dynamic flow[1] of the kind described in the introduction. Let t_k be the traversal time of the chain C_k. Then the problem would be simply:

$$(5) \qquad \text{maximize} \sum_{k=1}^{m} x_k(T + 1 - t_k)$$

subject to the constraints (4), since for $t_k \leq T + 1$, the coefficient $T + 1 - t_k$ is the number of times C_k can be used in the T periods, and for $t_k > T + 1$, a solution which maximizes assigns $x_k = 0$.

Now problem (4) – (5) can be put back in a form similar to that of (1) – (2), i.e.,

$$(6) \qquad \text{maximize } (T + 1) \sum_{j} (x_{0j} - x_{j0}) - \sum_{i,j} t_{ij}x_{ij}$$

subject to (1). It is not difficult to see that this may be restated as a capacitated Hitchcock transportation problem, for which several simple computational methods have been proposed [1, 2, 8].

To recapitulate, if one could establish the existence of a maximal dynamic flow of the simple kind previously described, the dynamic problem would not be difficult computationally.

[1] We shall give no formal definition of a dynamic flow in N, since the algorithm of the paper deals essentially only with static flows.

Encouraged by the fact that one of the authors (L. R. Ford, Jr.) had constructed the existence proof needed on the assumption that T was sufficiently large, we were led to an examination of the primal-dual method [3, 8] for solving the problem of finding a maximal static flow which minimizes total flow time $\sum_{i,j} t_{ij}x_{ij}$ over the class of all maximal static flows, i.e., of solving (1) and (6) for large T. Surprisingly, it turned out that the algorithm was grinding out, sequentially, maximal dynamic flows for T = 0, 1, 2, ..., and that the proof of this fact could be given directly from the algorithm itself.

3. Dynamic Algorithm. Routine I of the algorithm is an iterative process which constructs an integral static flow (x_{ij}), together with nonnegative integers π_i, one for each node P_i, having the properties

(7) a. $\pi_0 = 0, \quad \pi_n = T + 1$

b. $\pi_i + t_{ij} > \pi_j \longrightarrow x_{ij} = 0$

c. $\pi_i + t_{ij} < \pi_j \longrightarrow x_{ij} = c_{ij}$

d. $\pi_i + t_{ij} = \pi_j \longrightarrow 0 \leq x_{ij} \leq c_{ij}$.

To state the routine, we suppose we have an integral flow (x_{ij}) and node integers π_i satisfying (7) with $\pi_n = t$, and wish to construct (x'_{ij}) and π'_i satisfying (7) with $\pi'_n = t + 1$. To start out, one may take all $x_{ij} = 0$, all

$\pi_1 = 0.$[2]

Links P_iP_j for which $\pi_i + t_{ij} = \pi_j$ are called admissible below. Notice that at most one of P_iP_j, P_jP_i will be admissible, and that initially no links are admissible.

Routine I.

(a) Label P_0 with (P_n^+, ∞); consider P_0 as unscanned.

(b) Take any labeled, unscanned node P_i (initially P_0 will be the only such node); suppose labeled $(P_k^{\pm}, h)$. To all nodes P_j which are unlabeled and such that P_iP_j is admissible and $x_{ij} < c_{ij}$, assign the label $(P_i^+, \min (h, c_{ij} - x_{ij}))$. To all nodes P_j which are now unlabeled and such that P_jP_i is admissible and $x_{ji} > 0$, assign the label $(P_i^-, \min (h, x_{ji}))$. Consider P_i as scanned and newly labeled P_j (if any) as unscanned. Repeat until P_n is labeled or until no new labels are possible and P_n is unlabeled. In the former case go to (c) below; in the latter case, let (x'_{ij}) denote the flow, and proceed to (d).

(c) If P_n is labeled (P_k^+, h), replace x_{kn} by $x_{kn} + h$; if P_n is labeled (P_k^-, h), replace x_{nk} by $x_{nk} - h$. In either case, next turn attention to P_k. In general, if P_k is labeled (P_j^+, m), replace x_{jk} by $x_{jk} + h$, and if labeled (P_j^-, m), replace x_{kj} by $x_{kj} - h$, in either

[2] There are more efficient ways to start the routine, e.g., the procedure of [5] for finding a "shortest chain from P_0 to P_n may be used.

case turning attention then to P_j. Stop the replacement when P_0 is reached.[3] Starting with the new integral flow thus generated, discard the old labels and repeat (a) and (b) until the latter case of (b) obtains.

(d) Define π'_i by

$$\pi'_i = \begin{cases} \pi_i \text{ if } P_i \text{ is labeled,} \\ \pi_i + 1 \text{ if } P_i \text{ is unlabeled.} \end{cases}$$

Repeat routine I starting with (x'_{ij}) and π'_i (giving as new admissible links those P_iP_j for which $\pi'_i + t_{ij} = \pi'_j$) until the value of π_n has been boosted to T + 1. At this point a solution of the dynamic problem for T periods is at hand merely by decomposing the final flow obtained into chain flows. Routine II below uses a labeling procedure to effect this decomposition.

Routine II.

(a) Label P_0 with (P_n, ∞); consider P_0 as unscanned.

(b) Take any labeled, unscanned node P_i; suppose labeled (P_k, h). To all nodes P_j which are unlabeled and such that $x_{ij} > 0$, assign the label $(P_i, \min (h, x_{ij}))$. Consider P_i as scanned and newly labeled P_j (if any) as unscanned. Repeat until either P_n is labeled or

[3] The static flow value has been increased by h at this point.

new labels are impossible and P_n is unlabeled. In the former case, proceed to (c); in the latter case, stop.

(c) If P_n is labeled (P_k, h), replace x_{kn} by $x_{kn} - h$, next turning attention to P_k. In general, if P_k is labeled (P_j, m), replace x_{jk} by $x_{jk} - h$, and proceed to P_j. Stop the replacement when P_0 is reached.[4]

Repeat routine II until the latter case of (b) obtains; the flow (x_{ij}) has then been decomposed into chain flows.[5] The decomposition obtained in this way is of course not unique in general. However, any list of chain flows obtained from the final flow of routine I represents a solution to the dynamic problem for T periods. If the k-th chain of the list has a flow of x_k and a traversal time of t_k, then the maximal dynamic flow value is given by

$$\sum_k x_k(T + 1 - t_k) .$$

If one is interested only in this value and not the flow pattern, routine II of the algorithm may be ignored. Instead, let π_i be the final node numbers produced (i.e., $\pi_n = T + 1$), and define

$$\gamma_{ij} = \max (0, \pi_j - \pi_i - t_{ij}) . \tag{8}$$

[4] One sees that at the end of process (c) a chain carrying h units of flow has been traced out (in reverse).

[5] For an arbitrary flow, circulations and chain flows from P_n to P_0 may remain. This cannot happen, however, for a flow produced by routine I.

Then it can be shown that the maximal dynamic flow value is given by

$$(9) \qquad \sum_{i,j} \gamma_{ij} c_{ij} \ .$$

The node numbers π_i and link numbers γ_{ij} provide other significant information also. For example, links with positive γ_{ij} are "bottlenecks" for the dynamic flow — more specifically, a link P_iP_j for which $\gamma_{ij} > 0$ bottlenecks the flow in time periods $\pi_i + 1, \pi_i + 2, \ldots, \pi_i + \gamma_{ij}$.[6] Thus the flow value for T periods can not be increased unless some one of these links is given a larger capacity.

Notice that the dynamic algorithm has the feature alluded to in the introduction, namely, that in solving a problem for a given number of time periods T_0, optimal solutions for all fewer numbers of time periods $T < T_0$ can be obtained as by-products. In addition, it can be shown that at some stage $\bar{T}$ in the computation, a general solution of the problem for all numbers of time periods $T \geq \bar{T}$ is obtained.[7] (The value $\bar{T}$ is never greater than the maximal traversal time from P_0 to P_n, and is usually much less.) At the stage $\bar{T}$, the static flow obtained is a maximal flow which minimizes total flow time $\sum_{i,j} t_{ij}x_{ij}$ over all maximal static flows, and the links which lead from the final labeled set of nodes to its complement,

[6] See the definition of the minimal dynamic cut in [9].

[7] Thus the solution of the problem for all T is a finite process.

the unlabeled set, are a minimal (static) cut [4, 6, 7] in the network.

REFERENCES

1. Dantzig, G. B., "Application of the Simplex Method to a Transportation Problem," Activity Analysis of Production and Allocation, Cowles Commission Monograph 13, edited by T. C. Koopmans, John Wiley and Sons, New York, 1951, pp. 359-373.

2. Dantzig, G. B., "Recent Advances in Linear Programming," Management Science, Vol. 2, No. 2, January, 1956, pp. 131-144.

3. Dantzig, G. B., L. R. Ford, Jr., and D. R. Fulkerson, "A Primal-Dual Algorithm for Linear Programs," Linear Inequalities and Related Systems, Annals of Mathematics Study No. 38, edited by H. W. Kuhn and A. W. Tucker, Princeton University Press, 1956, pp. 171-181.

4. Dantzig, G. B., and D. R. Fulkerson, "On the Max Flow Min Cut Theorem of Networks," Linear Inequalities and Related Systems, Annals of Mathematics Study No. 38, edited by H. W. Kuhn and A. W. Tucker, Princeton University Press, 1956, pp. 215-221.

5. Ford, L. R.,(Jr.), Network Flow Theory, The RAND Corporation, Paper P-923, August 14, 1956.

6. Ford, L. R. (Jr.), and D. R. Fulkerson, "Maximal Flow through a Network," Canadian Journal of Mathematics, Vol. 8, No. 3, 1956, pp. 399-404.

7. Ford, L. R. (Jr.), and D. R. Fulkerson, "A Simple Algorithm for Finding Maximal Network Flows and an Application to the Hitchcock Problem," to appear in Canadian Journal of Mathematics.

8. Ford, L. R. (Jr.), and D. R. Fulkerson,"A Primal-Dual Algorithm for Capacitated Transportation Problems," to appear in Naval Logistics Research Quarterly.

9. Ford, L. R. (Jr.), and D. R. Fulkerson, Dynamic Network Flow, The RAND Corporation, Paper P-967, October 24, 1956.